

Transaction Processing (within an online context)

What is a Transaction?

- A set of actions that must either all be completed or non be completed.
 - If the transaction completed, the transaction is “Committed”
 - If the transaction does not complete, all state must be “Rolled-back” to the state of the system prior to the transaction

An Example Transaction

- Money Transfer

Move money from a checking account to a savings account.

To do this, one would:

- Remove (debit) money from the checking account
- Add (deposit) money into the savings account
 - » Either both of these steps need to occur or neither should occur.

- Purchase Tickets from “Ticketmaster.com”:

Transaction (1)

- Identify/reserve seats (seats in a “reserved state” for xx minutes)

Transaction (2)

- Validate credit card charge
- Remove seats from ‘available list’

Transactional systems

- A more formal / Exact definition of a transaction is that is must be able to pass the “ACID” test
- Requirements: ACID
 - Atomicity - the transaction must execute completely, or not at all
 - Consistency - Resources must remain consistent (ex. Same amount of money debited and credited)
 - Isolation - each transaction must appear to occur before or after any other transaction
 - Durability - If the transaction completes successfully, the state (changes due to the transaction) will survive any future failures

Characteristics of a transactional systems

Read Only vs Write (update)

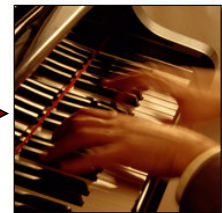
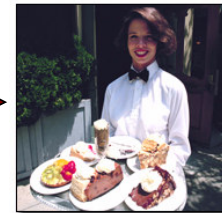
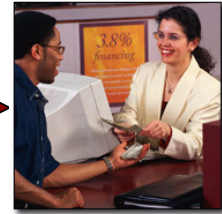
- Updates must lock the resource.

Because transactions can lock resources, they should be:

- **Short** (because they are locking resources)
- **Small** in scope (update a row in a database, not the entire database)



- **Many Users**
- **Repetitive**
- **Short Interactions**
- **Shared Data**
- **Data Integrity**
- **Low**
- **Cost/Transaction**



Deadlock!

- When transaction “B” needs to access a resource (ex. Database table) that is locked by transaction “A” (due to transaction “A” updating that resource)

and

Transaction “A” needs to access a resource that is locked by transaction “B”

- There is deadlock:
 - » Transaction “A” is waiting for Transaction “B” and
 - » Transaction “B” is waiting for Transaction “A”
- *To resolve, the system needs to cancel one of the transactions (and retry the canceled transaction “later”)*

Reviewing Batch vs Online Processing

Reviewing Batch processing

- There are two different ways to process input requests: batch processing and online processing.
- In *batch processing*, similar requests are saved up and submitted to the computer as a group. After they are submitted, the requests are processed sequentially. After all of the requests have been processed, the results are transmitted.
- Batch processing is like a warehouse manager saving up all the orders for the day, sorting them into types of order, and then filling the orders, for example, packaging the merchandise for shipping.

Online processing

- In *online processing*, requests are submitted and processed individually as soon as the computer receives them. Results are transmitted to a user or file as soon as the results are available.
- Unlike batch input, online input comes from multiple sources, and there is no way to sequence, predict or control the order and type of input. In the warehouse example, online processing is like the warehouse manager taking each order as it comes in and filling it immediately.

OLTP - Online Transaction Processing

- An OLTP is a class of software that administers transaction-related programs.
- OLTP software functions include:
 - Managing the user interface
 - Retrieving and modifying data
 - Tracking data locations and uses
 - Handling communications
 - Providing support functions for resource definition and use
 - Interfacing with security software

CICS

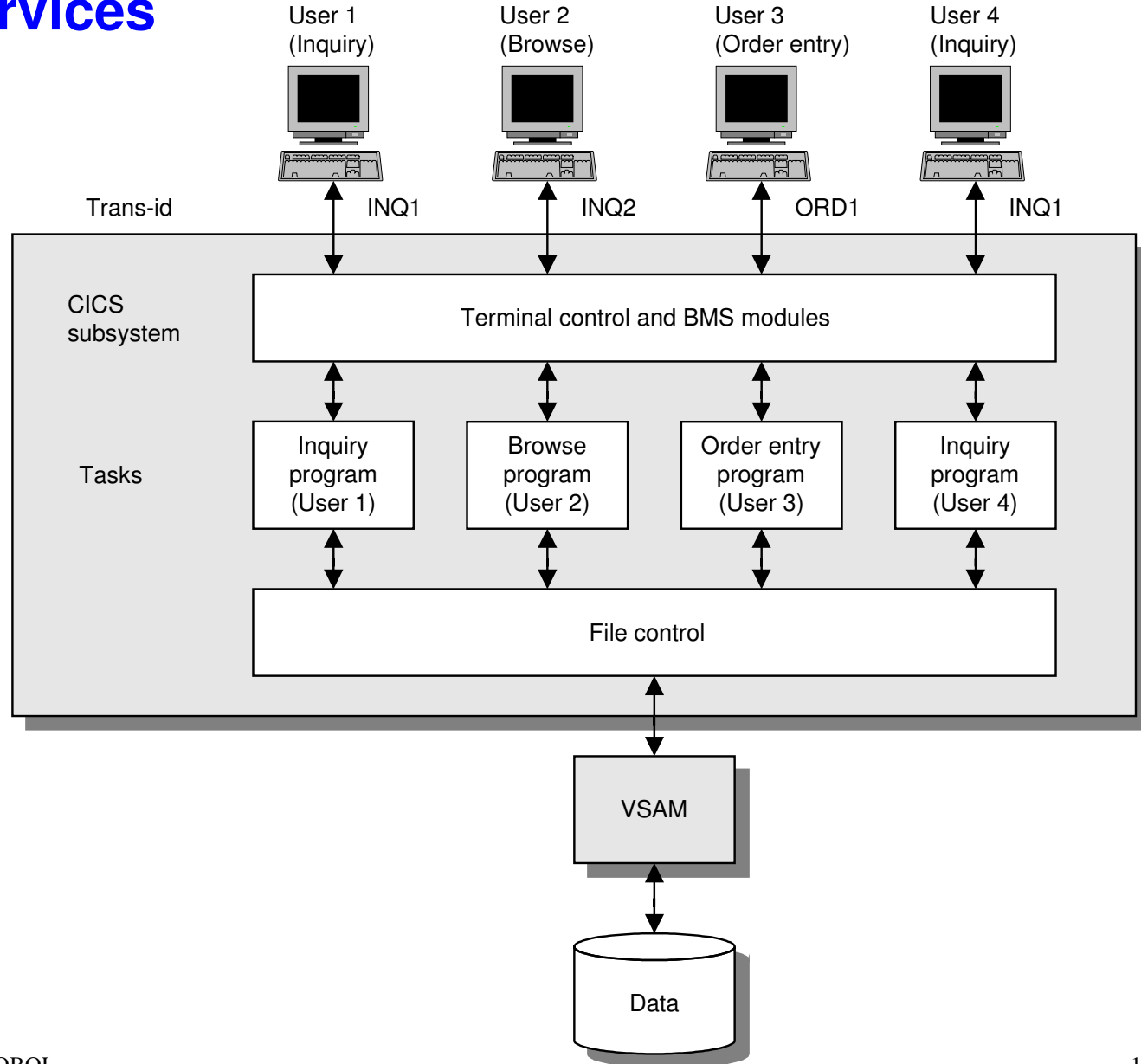
What is CICS?

- Customer Information Control System
- CICS is an OLTP product family from IBM.
- Transactional subsystem of z/OS which:
 - Runs online applications
 - Supports many users running the same application(s)
 - Manages the sharing of resources
 - Ensures the integrity of data
 - Prioritizes the execution of transactions
 - Ensures fast response time

What is CICS?

- CICS Transaction Server is a:
 - A Transaction Manager
 - A Transaction Monitor
 - An Application Server
- Used extensively:
 - Used for 30 Billion transactions / day
 - Used by 490 of IBM's top 500 customers
 - Close to 1,000,000 CICS programmers
- Systems that benefit from CICS include:
 - Bank ATM transaction processing services
 - Online library catalogues
 - Airline reservation systems

CICS services



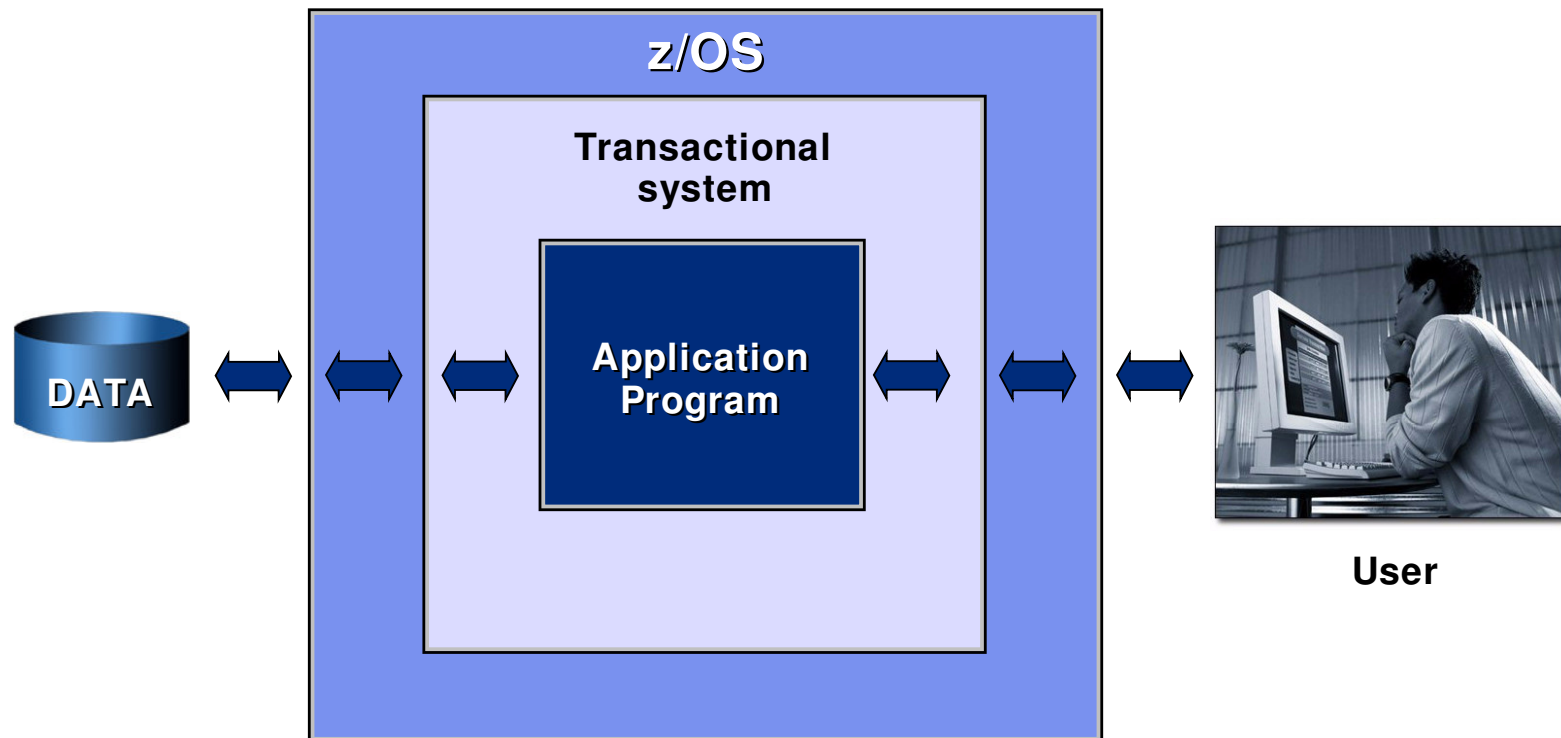
CICS functions

- Transaction processing systems perform interactive electronic commerce from a network of terminals, processing both inquiries and updates to data stored in databases.
- CICS is a general-purpose data communication system that can support the development of transaction processing applications in an OS/390 environment.
- Operating systems are designed to make the best use of a computer's resources. CICS helps by behaving as a middle layer, separating online application programs from other programs and handling their administration itself.

CICS functions

- Unlike batch programs, CICS programs do not make direct calls to the operating system. **Instead, CICS programs issue commands to perform terminal I/O, file I/O, program control, and other functions requiring a system resource.**
- In this way, CICS TS behaves as a mini-operating system within the actual operating system to provide an environment for CICS program execution.
- All CICS programs run in the CICS region of storage, under CICS control. CICS programs use CICS for all interfaces. CICS, in turn, interfaces with the Z/OS operating system.

CICS in a z/OS system



Languages & Platforms

- Languages:
- COBOL
- OO COBOL
- C
- C++
- JAVA
- PL/I
- Assembler

Platforms:

zSeries (z/OS, OS/390,VSE)

Intel servers

**TXSeries (AIX, HP-UX,
Solaris and Windows)**

CICS services for Application Programs

Application program interface: use CICS commands

Terminal control services: use Basic Mapping Support (BMS)

File & database control services:

- CICS file control (mainly VSAM)
- Database control (DL/I & DB2)

Other CICS Services: Task Control - Program Control - Temporary Storage (TS) & Transient Data Control (TD) - Interval Control - Storage Control - Dump & Trace Control

CICS —transactions

- In CICS, a *transaction* is a sequence of related operations or steps that together perform a specific function.
- Transactions might involve a single action or a set of operations, such as the set of steps in database updating, or the entire process of registering a new student in a college course.
- The steps for a student registration transaction would include:
 - 1) Reading the user's registration request
 - 2) Checking the current enrollment in the course
 - 3) Determining whether there is space for an additional student
 - 4) Adding the student to the class list
 - 5) Sending confirmation of the transaction to the user

CICS —Tasks

- In CICS, an instance of a particular transaction request by a computer operator or user is called a *task*. When a user invokes a transaction, CICS begins a task for that request. CICS also loads any application programs required for the transaction.
- For example, when a student registration request comes into the CICS system, CICS represents and keeps track of that request and its associated work by starting a unique task for it. CICS then loads the application programs that are required for executing that task.

CICS —tasks

- **CICS provides concurrent transaction processing**, which means that many users can enter and process requests at the same time. In order to allow for many users while ensuring swift response times, CICS employs multitasking methods.
- **Under CICS, all users share application programs and data files.** This means that if one transaction is being processed and another user make a similar request, CICS does not reload the application program. Instead, CICS starts a new task for the second request, using the same program or data file. CICS runs each task individually, briefly giving each task control of the CPU.

CICS —tasks

- Because concurrent users all share the same data files, if one user updates a database, the changes are available to all users immediately.
- This has obvious advantages in business applications such as airline reservation booking systems, where all system users need to be kept informed of factors like the current state of seat availability.

CICS —tasks

These steps are involved in processing a transaction:

- 1) **Entry** – A transaction ID (TRANSID) enters the CICS system.
- 2) **Task creation** – CICS creates a task to process the transaction. The task is now ready to be run.
- 3) **Dispatch** – CICS determines which of the ready tasks should be run next, and dispatches that task to be started.
- 4) **Execution** – The task invokes the appropriate CICS program and runs.

CICS —tasks

- 5) **Processing** – When the invoked program calls CICS to perform a service on its behalf, the task gives up control of the CPU and waits for the requested service to be completed.
- 6) **Redispatch** – After the requested service has been completed, the task is ready to run again, and CICS dispatches it again.
- 7) **Return** – When all work required to process the TRANSID is done, the program issues a RETURN command to return control to CICS.
- 8) **Termination** – CICS removes the task from the system.

CICS - Data integrity

- Protecting data from accidental loss or corruption is a primary concern of many businesses.
- CICS helps ensure the integrity of transaction-related data by following four key principles in transaction processing, known as ACID:

Atomicity

- Each transaction is treated as a separate unit.
- Either the entire transaction is committed (succeeds), or rolled back (fails).
- If a transaction fails, data cannot be corrupted; if it succeeds, the changes to the database are permanent.

CICS - Data integrity

Consistency

A transaction follows established protocols each time it is run.

The transaction uses data in the same way each time, which prevents data corruption across databases.

Isolation

No two transactions operate simultaneously on the same data, because CICS keeps tasks independent of one another until completed.

Durability

Committed data is saved by the system so that, even in the event of a failure and system restart, processing results are available.